Requested Patent: JP9106366A

AD

Title:

SYSTEM AND METHOD FOR FILE SYSTEM LOCK AND REPAIR FOR A COMPUTER OPERATING SYSTEM ;

Abstracted Patent: EP0759592, A3, B1 ;

Publication Date: 1997-02-26 ;

Inventor(s): SENATOR STEVEN T (US) ;

Applicant(s): SUN MICROSYSTEMS INC (US) ;

Application Number: EP19960650033 19960801 ;

Priority Number(s): US19950516429 19950817 ;

IPC Classification: G06F11/00 ;

Equivalents: DE69602500D, DE69602500T, US5765151 ;

ABSTRACT:

A system and method for file system fix-on-panic for a computer operating system which comprises an enhancement to the UNIX File System (UFS) that increases total system availability by detecting file system errors and determining whether on-line repair is possible and then locking, repairing and unlocking the failed file system. Availability of the entire computer system is increased since the mean time to failure for independent threads is increased by the amount of time up to the next failure and the mean time to repair for dependent threads is reduced to only the amount of time necessary for the repair. The system and method disclosed allows for repairs to be made during use, may be called from the user level and allows for blocking of only particular threads.

(19)日本国特許庁（ＪＰ）　(12)公開特許公報(A)　(11)特許出願公開番号

特開平9－106366

(43)公開日　平成9年(1997)4月22日

(51)Int.Cl.*	識別記号	庁内整理番号	FI		技術表示箇所
G06F 12/00	531		G06F 12/00	531Z	
				531R	

審査請求　未請求　請求項の数18　OL　（全 12 頁）

(21)出願番号　特願平8－217475

(22)出願日　平成8年(1996)8月19日

(31)優先権主張番号　516429

(32)優先日　1995年8月17日

(33)優先権主張国　米国（US）

(71)出願人　591064003
サン・マイクロシステムズ・インコーポレーテッド
SUN MICROSYSTEMS, INCORPORATED
アメリカ合衆国　94043　カリフォルニア州・マウンテンビュー・ガルシア　アヴェニュウ・2550

(72)発明者　スティーヴン・ティー・セナター
アメリカ合衆国コロラド州80920，コロラド・スプリングス，ウエストミンスター・ドライヴ　8625

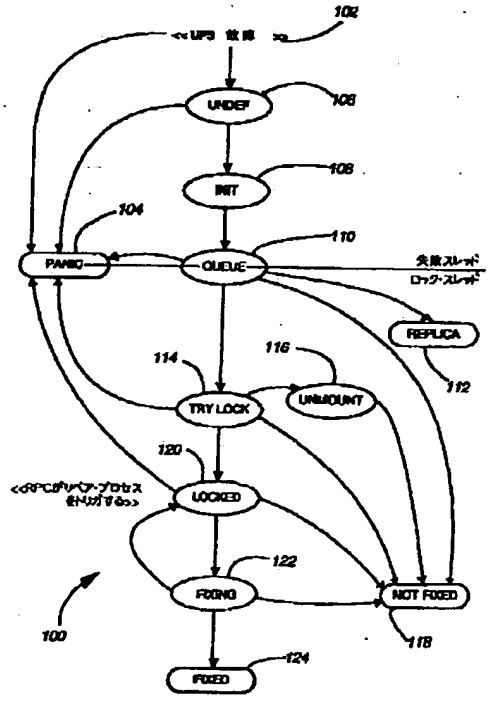
(74)代理人　弁理士　湯浅　恭三　（外５名）

(54)【発明の名称】　コンピューター・オペレーティングシステムのためのファイルシステム・パニック回復のシステムと方法

(57)【要約】

【目的】　ユニックス・ファイルシステムのファイルシステム・パニック回復のためのシステム及び方法を実現して全システムの可用性を増加する。

【構成】　ユニックス・ファイルシステム「UFS」に対する強化から成り、ファイルシステム・エラーを検出して、オンライン修理ができるかどうか決め、それから故障したファイルシステムをロックし、修復して、アンロックすることによって、全システム可用性を増加する、コンピューター・オペレーティング・システムに対する、ファイルシステム・パニック回復のためのシステム及び方法である。次の故障までの時間と従属するスレッドの平均修復時間が修理に必要な時間だけに減少することによって、独立のスレッドの故障までの平均時間が増加するので、全計算機システムの可用性が増加する。開示されたシステムと方法は、使用中に修理することを許し、ユーザーレベルから呼ぶことができて、特定のスレッドだけのブロッキングを許す。



【特許請求の範囲】

【請求項１】　コンピューター・オペレーティング・システムの故障したファイルシステムを選択的に誤りロックする方法であって、以下のステップから成っている：前記コンピューター・オペレーティング・システムの故障したファイルシステムにおけるエラーを検出するステップ；前記故障したファイルシステムのオンライン修復ができるかどうか決めるステップ；前記オンライン修復ができるならば、故障したファイルシステムをロックするステップ；前記故障したファイルシステムの前記オンライン修復を成し遂げるステップ；前記故障したファイルシステムをアンロックするステップ。

【請求項２】　請求項１に記載の方法において、前記検出するステップが、次のステップからなる：前記故障したファイルシステムの矛盾に注目するステップ；前記故障したファイルシステム矛盾に対応する故障レコードをつくるステップ；オペレーティング・システム・ロック・スレッド上に前記故障レコードを置くステップ。

【請求項３】　請求項２に記載の方法において、前記矛盾が、オペレーティング・システム故障スレッドに応答して注目されることを特徴とする方法。

【請求項４】　請求項１に記載の方法において、前記ロッキングのステップが、次のステップから成る：関連するコンピューターの大容量記憶装置の上で、前記故障したファイルシステムと関連するデータブロックをアクセス不可能と記録するステップ。

【請求項５】　請求項１に記載の方法において、前記オペレーティング・システムが、前記故障したファイルシステムのオンライン修復ができるかどうか決めるステップが不可能であることに応答してシャットダウンされることを特徴とする方法。

【請求項６】　請求項１に記載の方法において、更に次のステップを含む：前記故障したファイルシステムがロックされる一方、他のオペレーティング・システム・ファイルシステムに対応している他のスレッドが継続するのを許すステップ。

【請求項７】　以下の要素からなるコンピューター・プログラム・プロダクト：コンピューター・オペレーティング・システムの故障したファイルシステムを選択的に誤りロックするためにコンピューター可読のコードを記録されたコンピューターが使用できる媒体；コンピューターに、前記コンピューター・オペレーティング・システムの故障したファイルシステムにおけるエラーの検出をさせるようにするために構成されたコンピューター可読のプログラムコード・デバイス；コンピューターに、前記故障したファイルシステムのオンライン修復ができるかどうかの決定をさせるようにするために構成されたコンピューター可読のプログラムコード・デバイス；コンピューターに、前記オンライン修復が可能ならば前記故障したファイルシステムのロッキングをさせるようにするために構成されたコンピューター可読のプログラムコード・デバイス；コンピューターに、前記故障したファイルシステムのオンライン修復をさせるようにするために構成されたコンピューター可読のプログラムコード・デバイス；コンピューターに、前記故障したファイルシステムをアンロックさせるために構成されたコンピューター可読のプログラムコード・デバイス。

【請求項８】　請求項７に記載のコンピューター・プログラム・プロダクトにおいて、前記コンピューターにエラーの検出をさせるようにするために構成された前記コンピューター可読のプログラムコード・デバイスが更に以下の要素からなる：コンピューターに、前記故障したファイルシステムにおける矛盾に注目することをさせるために構成されたコンピューター可読のプログラムコード・デバイス；コンピューターに、前記故障したファイルシステムの矛盾に対応している故障レコードをつくることをさせるようにするために構成されたコンピューター可読のプログラムコード・デバイス；コンピューターに、オペレーティング・システム・ロック・スレッド上に前記故障レコードを置くことをさせるようにするために構成されたコンピューター可読のプログラムコード・デバイス。

【請求項９】　請求項７に記載のコンピューター・プログラム・プロダクトにおいて、前記コンピューターにロッキングをさせるようにするために構成されたコンピューター可読のプログラムコード・デバイスが更に以下の要素からなる：コンピューターに、関連するコンピューター大容量記憶装置の上で、前記故障したファイルシステムと関連するデータブロックをアクセス不可能であると記録させるようにするために構成されたコンピューター可読のプログラムコード・デバイス。

【請求項１０】　請求項７に記載のコンピューター・プログラム・プロダクトにおいて、更に次の要素を含む：コンピューターに、前記故障したファイルシステムのオンライン修復ができないという決定に答えて前記オペレーティング・システムをシャットダウンさせるようにするために構成されたコンピューター可読のプログラムコード・デバイス。

【請求項１１】　請求項７に記載のコンピューター・プログラム・プロダクトにおいて、更に次の要素を含む：コンピューターに、前記故障したファイルシステムをロックする一方、他のオペレーティング・システム・ファイルシステムに対応している他のスレッドの続行を許させるようにするために構成されたコンピューター可読のプログラムコード・デバイス。

【請求項１２】　アプリケーションプログラムを実行するために、その上にロード可能なコンピューター・オペレーティング・システムを含むコンピューターにおいて、前記オペレーティング・システムは、前記アプリケーションプログラムからアクセス可能な複数のファイル

システムを持っており、次の要素から成る：前記複数のファイルシステムの内の故障したものにおけるエラーを示すための故障スレッド；前記故障スレッドに応答して始まったロック・スレッド；もしそれについてオンライン修復ができるなら、前記故障したファイルシステムを選択的にロックするために前記ロック・スレッドに応答する誤りロック；オンライン修復が実行されているとき、前記誤りロックに応答して前記複数のファイルシステムの内の前記故障したものに他のスレッドがアクセスするを禁止し、これによって前記故障したファイルシステムのオンライン修復の実施に応答して削除されるブロック。

【請求項１３】　請求項１２に記載のコンピューターにおいて、前記オペレーティング・システムが次の要素から成る：前記アプリケーションプログラムに前記オペレーティング・システムをインターフェースするためのユーザ層；前記ユーザ層の基礎をなしているカーネル層；前記カーネル層の基礎をなしていて、それらの間でファイルシステム・インタフェースを定義しているファイルシステム層。

【請求項１４】　請求項１２に記載のコンピューターにおいて、前記ブロックが、前記ファイルシステム・インタフェースで実現されることを特徴とするコンピュータ。

【請求項１５】　請求項１３に記載のコンピューターにおいて、前記ファイルシステム層が、ユニックス・ファイルシステム層から成ることを特徴とするコンピュータ。

【請求項１６】　請求項１３に記載のコンピューターにおいて、前記ファイルシステム・インタフェースが、VFS層から成ることを特徴とするコンピュータ。

【請求項１７】　請求項１２に記載のコンピューターにおいて、更に次の要素を含む：前記複数のファイルシステムと関連するデータのブロックを記憶するための、前記コンピューターと関連するコンピューター大容量記憶装置。

【請求項１８】　請求項１７に記載のコンピューターにおいて、前記故障したファイルシステムと関連する前記コンピューター大容量記憶装置上の選ばれたデータのブロックが、前記ロック・スレッドに応答してアクセス不可能であると記録されることを特徴とするコンピュータ。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】一般に本発明は、コンピューター・オペレーティングシステム「OS」のファイルシステム「FS」の分野に関連がある。特に、本発明は、ファイルシステム「パニック回復」に対する「誤りロック」を選択的にインプリメントし、削除する方法およびシステムに関連する。これは、ユニックス・システムVファイルシステム（「VFS」、同様に「VFSop」又は「Vnode」として名を知られている）階層又は同様のOSファイルシステム・インターフェースを含んでいるコンピューター・オペレーティングシステムに関する特別なユーティリティである。

【０００２】

【従来の技術】

【０００３】

【発明が解決しようとする課題】コンピューターの大量記憶媒体上の名をつけられたオブジェクトを記憶したものであるようなファイルのグループ化として、ファイルシステムは定義できる。このように、ファイル・システムは、そのファイルシステムが適当に作用するために、お互いが全く一致していなければならない一組のメモリー上とディスク上のデータ構造を保守する。これらのデータ構造が一貫していないとき、例えばユニックス・オペレーティングシステムは、強制的なトータルシステムしゃ断「パニック」を実施する。それから、この「パニック」は、コンピュータ・システムをシャットダウンし、全てのディスク上の矛盾の修復と正常な系統運用のリジュームをするために必要な時間の間、全部のシステムを利用できなくする。

【０００４】ユニックスへのロックＦＳ「lockfs」機能性の追加を通して、任意のファイルシステムのために一般的にインプリメントされてもよい全ファイルシステムをロックする能力が、提示された。この特長は、ロックの違なる種類に従って選ばれたオペレーションを抑止する能力を提供する。例えば、ファイルシステムは、抑止されていた他のいかなるオペレーションもなく、一時的かつ選択的に変更不可能にされることができる（例えば、ここでファイルが、削除されるのを妨げられることができる）。「抑止された」オペレーションは、ファイルシステムが再びロックを解かれるまで単に待つだけものである。この点について、「削除ロック」又は「ネーム・ロック」は、ファイルシステムが、依然として、ほんのある程度の定義済み方法だけにおいて変わることができる一種のロックである。ロックされたファイルシステムへのアクセスは、そのファイルシステムのロックが解除されるまでの認め得るディレイを生ずる。

【０００５】これに対して、「ハード・ロック」は特定のファイルシステムが単に更なるアクセスから除外されるものである。例えば、記憶媒体欠陥が検出され、データ・リカバリ動作が実現されなければならないとき、ハード・ロックは起動される。ハード・ロックは、ネームスペースからファイルシステムを削除しない、しかし、エラーにより失敗したその内部の何れへもアクセスし、そしてハード・ロックをクリアする方法は、ファイルシステムをアンマウントすることだけである。特に、与えられたコンピューター上のファイルシステムが整合性問題を持っていれば、コンピューターは「ダウン」し、そ

れが提供するサービスの全てが、全てのユーザーから利用できなくなる。この整合性問題に遭遇するのが、偶然ネットワーク上のサーバ・コンピューターであるとき、これは、既存のクライアント-サーバ計算機システムにおいて特に緊急で高価な状況である。

【０００６】システム故障時間を避ける試みにおいて、ファイルシステムそれ自身にチェック・アルゴリズムを埋めこむことは、以前に提案された。この手法の典型は、チェックが二相コミット・プロトコルを利用しているコアにおいてされる、ヴェリタス（Veritas）からのVXFSソフトウェアと、よりデータベース風のモデルに構築したIBMジャーナルド・ファイルシステム「JFS」である。いずれにせよ、チェック機能をインプリメントするためにファイルシステムに追加した、追加の中央処理装置（ＣＰＵ）「オーバーヘッド」が、全体的により遅いシステム動作をもたらした。アウスペックス（Auspex）によって実現されたもう一つのチェックのアプローチは、単にファイルシステム・オペレーションを提供するだけに特化されたＣＰＵ上で動作するアルゴリズムを利用する。従って結果として生じるシステムは、それが設計された特別なハードウェア・インプリメンテーションに特有であり、したがって他のアーキテクチャに対して限られた適用可能性しか備えていない。

【０００７】

【課題を解決するための手段】本発明のシステムと方法は、ファイルシステム「パニック回復」を実現するために用いられる「誤りロック」と命名される代替ファイルシステム・ロッキング機構を既存の「書き込み」及び「ハード」ロックに提供する。本明細書において開示されたように、誤りロックは（ハード・ロックのように）すべてをロックするが、ファイルシステムが一貫させられたあと、（書込みロックのように）依然としてロックを解除されることができる。本発明のシステムと方法を利用すれば、エラーに遭遇した特定のファイルシステムと、そのファイルシステム・サービスを用いていた特定のユーザーが影響されるだけで、これにより独立したユーザーにとっての全システムの可用性が大いに増加する。

【０００８】本発明のファイルシステム・パニック回復システムと方法は、ユニックス・ファイルシステム「UFS」を拡張して、ファイルシステム・エラーを検出して、オンライン修復ができるかどうか決め、それから故障したファイルシステムをロックして、修復し、そしてアンロックすることによって全計算機システム可用性を増やすために役に立つ、特別なユーティリティである。本明細書において明らかにしたシステムと方法は、VFS層またはその同等物、任意のシステムVベースのユニックス・システム、IBMのAIXまたはマイクロソフトのNTオペレーティングシステムを含んでいるオペレーティングシステムの上で、同様に有利に実現することができる。

【０００９】本発明のシステムと方法が、工夫されたあるオンライン・ファイルシステム・チェックへのアプローチを提供する既存のロック機構の付属物は、モノリシックの上でカーネルを実現して、マルチプロセッサー上の実行であることができる。それは、使用中にファイルシステム修繕でき、ユーザーレベルから呼ぶことができ、そのうえUFSロックファイル・システム「lockfs」層に、一般のファイルシステム・データ構造とアルゴリズムに関して非侵入的であると同時に特定のスレッドだけを抑止させるように修正することができる。

【００１０】本明細書において明らかにしたようなコンピューター・オペレーティングシステムの故障したファイルシステムを選択的にエラーロックするための特定の方法のインプリメンテーションとその方法を実現するコンピューター・プログラム・プロダクトにおいて、この方法は、コンピューター・オペレーティングシステムの故障したファイルシステムにおけるエラー検出し、故障したファイルシステムのオンライン修復ができるかどうか決めるステップから成る。この方法は更に、オンラインで修復できるなら故障したファイルシステムをロックし、故障したファイルシステムのオンライン修復を成し遂げて、故障したファイルシステムをアンロックするステップから成る。より特定のインプリメンテーションにおいて、この検出するステップは、故障したファイルシステムにおける矛盾に注目し、故障したファイルシステムの矛盾に対応する故障レコードをつくって、その故障レコードをオペレーティングシステムのロック・スレッドに置くステップによって遂行される。

【００１１】同様に、アプリケーションプログラムからアクセス可能な複数のファイルシステムを持っているオペレーティングシステムにより、アプリケーションプログラムを走らせるのために、その上にロード可能なコンピューター・オペレーティングシステムを含むコンピューターを開示する。オペレーティングシステムは、複数のファイルシステムの内の故障したものにおけるエラーを示す故障スレッドと、故障スレッドに応答して開始されたロック・スレッドとを含む。それについて、オンライン修復ができるならば、誤りロックは故障したファイルシステムを選択的にロックするためのロック・スレッドに応答する。複数のファイルシステムの内の故障したもののオンライン修復が実行されている間、それに他のスレッドがアクセスすることを禁止するために、ブロックが誤りロックに応答し、その後、ブロックは故障したファイルシステムのオンライン修復の完成に応答して削除される。

【００１２】本発明の前記及び他の特長と目的が、そして、それらを達成する方法がより明白になるだろう。そして発明それ自身が添付の図面に関連してとられた好適な実施例の以下の記述の引用によって、最もよく理解できるだろう。

【００１３】
【実施例】本発明が用いられるシステム環境は、汎用コンピューター、ワークステーション・パソコンが、いろいろな型の通信リンクを通して、クライアント-サーバー構成において接続される、一般の分散された計算機システムを包含し、ここで（多くはオブジェクトの形の）プログラムとデータが、システムのいろいろなメンバによって実行のために利用可能であり、システムの他のメンバによってアクセス出来るようにされる。汎用ワークステーション・コンピューターの要素のいくつかが、図１において示される。ここで、入出力（I/O）セクション２、中央処理装置（ＣＰＵ）３、及びメモリ・セクション４を持っているプロセッサ１が示される。I/Oセクション２は、キーボード５、表示装置６、磁気ディスク装置９とコンパクトディスク・リードオンリーメモリ（CD ROM）駆動装置７に接続している。CDROM装置７は、典型的にプログラム１０とデータを含むCDROM媒体８を読みとることができる。本発明の装置と方法を成し遂げるために機構を含んでいるコンピューター・プログラム製品は、そのようなシステムのCDROM ８上、磁気ディスク装置９上、またはメモリー・セクション４に置くことができる。
【００１４】図２を参照すると、図１において記載された複数のコンピューターのネットワークから成っている計算機システムでのアプリケーション・プログラムの実行の基礎をなしている妥当なユニックス・オペレーティングシステム層２０の簡略概念上の図解が示される。図示された詳細は、「ユーザ」（システム・コール）層２２、「カーネル」２４と「UFS」層２６である。カーネル２４内部には、UFSモジュール２８と、いくらかの例としてネットワークファイルシステム「NFS」モジュール３０が存在する。UFSモジュール２８は、例えばファイル命名、記憶領域割付けと他のサービスを含むことができる。図示のように、（Op１、Op２およびOp３のような）たくさんのオペレーションがVFS層３２でUFSモジュール２８とNFSモジュール３０に生じることができる。（VFSopまたは Vnode層としても知られている）VFS層３２は、いろいろなオペレーションを実行すｒことができる、カーネル２４とUFS層２６の間のインターフェースである。
【００１５】図３を参照すると、以前に図２において記載したVFS層３２を含む、（ユーザー層２２、カーネル２４とUFS層２６から成る）システム層２０が示される。それに加えて、記憶装置４０が、たくさんのUFSスレッド３８とともに示される。UFSスレッド３８は、実行しているプログラムまたはプログラムコードを実行することができる独立の「もの」として都合よく考えることができる。（これは複数のスレッドを持つことができる独立のものである「プロセス」との対比である）。
【００１６】記載した正常動作において、特定のファイルシステム・オペレーションを要求するアプリケーションプログラムの実行に応じて、UFSスレッド３８がユーザー層２２、カーネル２４とVFS層３２を介してUFS層２６に通る。さらに図４を参照すると、ファイルシステム故障の発生時の以前の図のシステム層２０と記憶装置４０が示される。この点について、以下の定義を参照されたい：「Authoritative System Reference（命令的システム参照）」または「ASR」はシステムの修正挙動を生じる仮想のエンティティーである。
【００１７】「Error（エラー）」はＡＳＲによって生じたシステム挙動と実際のそれのと間の差である。
【００１８】「Fault（フォールト）」はエラーを起こす可能性を持つ原因である。「Failure（故障）」はフォールトの表明である。
【００１９】「UFS Failure」はその明確に示された動作からそれているUFSのインスタンスである。あるインプリメンテーションにおいて、UFS故障は、「synchronous failures（同期故障）」（プログラムコードが、明示的にチェックする）か、「asynchronous failures（非同期故障）」（トラップ処理機構によって取り扱われる）のいずれかである。UFS故障の後の型によって提示された問題は、本明細書において開示されたシステムと方法に適用できない。
【００２０】サン・マイクロシステムズ社（本発明の譲り受け人）によって開発され、ライセンスを与えられるソラリス・オペレーティングシステムの、特定のインプリメンテーション内では、同期故障の２つの型に遭遇することができる：「Assertions（アサーション）」または「 Asserts（主張）」はプリ・プロダクション・ソフトウェアにおいてだけチェックされる条件式である。真であるとき、システムは強制的にシャットダウンされる。
【００２１】「Panics（パニック）」は常に、チェックされる条件式である。それらは「永久的なアサーション」であり、語「panic（パニック）」は「強制されたシステム・シャットダウン」を意味するための名詞、そして「シャットダウンにシステムを強制する」ことを意味する動詞としてしばしば用いられる。
【００２２】「Failing Thread（故障スレッド）」はUFS故障が現れる状況におけるスレッドである。従属する（そして、独立の）スレッドが、故障ファイルシステムのリソースに依存する（またはしない）スレッドとしてそれぞれ、定義される得る。
【００２３】「Availability（可用性）」はシステムが受け入れられる応答時間以内にサービスを提供することができる予想された時間の部分である。重大な影響を伴わずに短期間サービスを与えず、または遅らせることができるシステムを記述することが出来る。それは量的に次のように定義できる：

$$Availability = \frac{MeanTimeToFail}{MeanTimeToFail + MeanTimeToRepair}$$

または代わりに：

$$Availability = \frac{1}{\left(1 + \frac{MeanTimeToRepair}{MeanTimeToFail}\right)}$$

本発明のシステムと方法は、システムのユーザーを故障したファイルシステムに従属するユーザーと従属しないユーザーとに分割して、可用性の粒状性を全システムからファイル・システム・サービスのそれへ減らすことを仮定している。

【００２４】表１：システム・コール故障セマンティクス：ローカルおよびリモート・スレッド

スレッド	システム・コール故障セマンティック
ローカル+リモート：故障、従属	リターン・エラー（ERESTART）
ローカル：従属	アンロック又はアンマウントまでブロック
リモート：従属	リターン・エラー（EWOULDBLOCK）
ローカル+リモート：独立	故障なし：停延なし

従来のUFSインプリメンテーションにおいて、多量の状態は、インコアで保持された。しかし、この状態の全てが、UFSそれ自身によって管理されるのではない。UFS故障は、このインコア状態の少なくともいくつかが信頼できないことを示す。これから回復するために、以下の動作が、必要である：

（１）故障されたファイルシステムは、静止されなければならない；

（２）不良インコア状態は、破棄されなければならない；

（３）ディスク上の状態は、少くとも一貫しているように、そして（多分）訂正するために検査されなければならない；

（４）インコア状態は、一貫した値から再初期化されなければならない；そして、

（５）ファイルシステム静止で「止まっていた」オペレーションは、進むために解除されなければならない。

【００２５】特にユニックス・オペレーティングシステムにおいて、この問題はpanic()機構によって言及される、そして、全てのインコア状態は破棄される。修復と再初期化は、再起動時に起こる。必要なデーモンが再開されるまで、プロトコルの性質によって、ネットワークファイルシステム・オペレーションは、停止している。破棄されたローカル操作は、消失する。fsckによってされる修復を除いてlockfsは上記の機能の全てを実行する。UFS故障は、UFS内部の多くの異なる論理レベルで遭遇されることができる。このように、どの複数または単数のリソースのロックが故障の時点で保持されているかに関する保証がない。これは、仮想計算機のようなUFSが依存する他のサブシステムと同様にUFSリソースとロックに対して真である。伝統的なpanic()ルーチンは、ロック獲得アルゴリズムにおける明示されたコードによって、この問題に対処する。パニックのときは常に、ロック要求は許可され、他の非パニック・スレッドは、実行することを防げられる。

【００２６】lockfsは、ioctl()システム・コールから起動される。したがって、それはより階層的にリソースとロックに関して作用する。しかし、特別なケースのコードをUFS lockfsとそれによって起動できる他の全てのルーチンに入れることは、追加の機構の採用を必要とし、一般的UFSデータ構造とアルゴリズムに強制する。

【００２７】UFSインプリメンテーションは、アロケーション・ビットマップと要約構造内部の冗長情報を含む。これの効果は、故障が実際のエラーが生じた時より非常に後の時間に生じる傾向があるということである。したがって、エラーと故障条件「パニック・ストリング」の間に多対１関係がある。したがって、提案された機構は、検出された全ての悪い構造を破棄し、そしてそれらを一貫した値に再初期化する。結果として、常にそれは故障に対処する際の現在のパニック・再起動メカニズムと同様である。

【００２８】故障の影響があまりに激しいとき、あるいは、エラーそれ自身が解決できるときは、修復を試みることは適当でない。これらの影響は、ＭＴＴＦかＭＴＴＲが受け入れ不可能に高レベルであるように表せる。他の表現をすれば、故障が、あまりにしばしば生じているか、あるいは、故障があまりにたくさんの被害を引き起こす。これは、一組の新しい調整可能の変数によって制御可能である。エラーが再現可能であるならば、エラーをマスクするために本発明のファイルシステム・パニック回復を用いるよりは、むしろバグをフィックスする方が正しい。本明細書において開示されたシステムのユーティリティと方法は、間欠的に生じて、まだ再現可能でないエラーに対処するためのものである。

【００２９】一般的なケースにおいて、UFSファイルシステムを訂正する試みが、なされるべきである。しかし、いくつかの特定のUFSファイルシステムは、それらに課された追加の制約を持つ。ルート「/」及びユーザ「/usr」ファイルシステムはfsckを起動する必要がある。結果として、それらをロックすることは、デッドロ

ック条件を引き起こす。活動中のカーネル・アカウントファイルが在り、あるいはスワッピングが生じることができるUFSファイルシステムも、同様のデッドロック条件をつくることができる。したがって、これらの特定のファイルシステムの上で、パニックは起動される。

【００３０】世代番号が変化したinodesは、lockfs調和アルゴリズムによって認識される。古いinodeは、それを強制的にアンマウントしたファイルシステム上のファイルの参照に現れさせるために修正される。新しいinodeモードまたはアルゴリズムは、必要でない。割当てinode（ファイルシステムの割当てファイルと関連させられたinode）が削除されるならば、割当て施行はこのファイルシステム（closedq()に類似した方法において）の上で使用禁止にされる、そして、警告メッセージは記録される。これが「fsck-p」のためのケースで決してなく、手動の修復がinodeが削除されるようにするために必要であることに注目されたい。アンリンクされたが、依然として参照されているファイルは検出可能で、削除されないだろう。Fsckの出力は、ロガー・コマンドによって獲得されて、syslogファイルに送られる。

【００３１】図４の典型的な例で示すように、ファイルシステム故障のイベントで、故障スレッド４２は、ユーザ層２２、カーネル２４を介してUFS層２６へ通る。故障スレッド４２の検出時に、参照番号４４で示されたように、UFS故障レコードがつくられ、ロック・スレッド４６の作業待ち行列の上に置かれる。故障が生じた時ロック・スレッド４６は始まり、そしてそれは、本明細書の下文に記述されるように、記憶装置４０上のブロックにその後はアクセスできないと記録して、そのファイルシステムをロックするように作用する。

【００３２】参照番号４８で示されるように、それから、故障スレッド４２は、エラーを持って復帰する。応答して、参照番号５０において、記憶装置４０で、故障したファイルシステムのスーパーブロックのfs_cleanフラグは不良と記録され、そして、参照番号５２によって示されるように、誤りロック３６が不良のファイルシステムの上に置かれる。示されるように、誤りロック３６は、VFS層３２で他のUFSスレッド５４を抑止するのに役立つ。参照番号５６で、lockfsが復帰する時、UFS故障は状態「LOCKED」へ遷移する。

【００３３】ファイルシステムの矛盾が検出されるとき、以前に記述されたプロセスが起動される。しかし、矛盾はその瞬間に起こることができず、そして、いつそれが生じたかを示す十分な情報がその時点で利用可能でないので、アプリケーション・チェッカーを必要とする。lockfsに関して、与えられたファイルシステムのための、全てのカーネル２４の状態（記憶ブロックと関連する「管理作業」と例えばファイル名）は、スタップを除いて破棄される。「アンロック」プロセスは調和と呼んだロック・ディスク機能をトリガーし、そして、記憶装置４０から現在の状態を検索する。そして、実際それは、それの後に残された以前に使用中だったもののための全てのスタップのために記憶装置４０から現在の状態を検索するだけである。したがって、ファイルが変化するならば、記憶装置４０上のファイルの状態は、正確だと仮定される。ファイルがいくらか削除されたならば、エラーは後に検出される、しかし、それは単なるアプリケーション・エラーであって、ファイルは単に再び書込まれる。lockfsによって可能化された調和は、このチェックを許す。

【００３４】この文脈において、図４は概念的に内部のデータ構造を図示する、しかし、問題があることを実際に識別するプロセスはエラーコード「erestart」を参照番号４８で提供する。このエラーコードは、いくつかのアプリケーションのケースにおいて、チェックがその特定のエラーコードに対して行われ、同じオペレーションが再びユーザに透過的に試みられるだろうことを意味する。それにもかかわらず、全てのオペレーションが再実行できる訳ではない。例えば、ファイルをつくろうとする際に、いくつかのケースにおいては、ファイルをつくることは、ファイルシステムが損害を与えられた仕方よって非常に難しい。

【００３５】さらに図５を参照すると、ロック・スレッド４６と関連する図４のシステム層２０が示され、修復プロセスを始めるために参照番号５８によって示されるように、RPCがつくられる。参照番号６０で、inetdはRPCの生成を通知され、参照番号６２で示されるように、UFSデーモンが始まる。応答において、fsckプロセスが参照番号６４で始まる。その間に、RPCが非同期だったので、参照番号６６で示されるように、inetdが復帰する。参照番号６８で、fsckが記憶装置４０上のスーパーブロックに修復するように記録し、そして、参照番号７０でUFS故障レコードが「修復中」状態へ遷移する。

【００３６】以前に記述されたように、ロック・スレッド４６は、修復プロセスを始める。それは、ファイルシステム一貫性チェッカーをカーネル２４から呼んで実行させ、それから、それが実際に始めることを確実にするようチェックする。言い換えると、ファイルシステムがエラーでロックされているならば、lockfsのために以前に記述されたように、記憶装置４０上に残されたある程度のキーがある。決してコンピューターをデッドロックさせないことは、ロック・スレッド４６のジョブの一部である。従って、所定の期間内に何も起こらないならば、それがエラーを発生させ、システム管理者がそれを単にサービスの遅延と見えるよりは、注意すべきものと認識するだろうから、コンピューターはシャットダウンされる。一貫性チェッカーを呼ぶことによって、システムは前進し、ファイルシステムを修復する。

【００３７】さらに図６を参照すると、参照番号７２で、fsckがファイルシステム修復を完了し、記憶装置４

0の上でファイルシステムに「クリーン」と記録する。それからFsckが、参照番号７４で同様にlockfs ioctlに「アンロック」を発行する。応答において、参照番号７６でlockfsサブモジュールが、UFS故障レコードに「修復済み」と記録するためにパニック回復ルーチンに呼ぶ。参照番号７８でlockfsは戻り、他のスレッド８０が、誤りロック３６（図４）の除去によってUFS層２６まで続く事を許される。

【００３８】図7を参照すると、以前に図3-6に関して記述された本発明の方法とシステムの特定のインプリメンテーションに従うUFS故障のイベントにおける複数の状態の間の可能な遷移１００を図示している状態図が示される。

【００３９】イベント１０２で、UFS故障が、検出された。Struct ufs_failure（又はtypedef uf _t）レコードが、各UFS故障と関連させられる。ファイルシステムが故障した時、ufsvfspから現在のstruct ufs_failureへのポインタが、ある。これらは、struct ufs_queueによって記述され、UFSスレッドによって用いられる待ち行列に置かれる。このスレッドは、待ち行列上に非ターミナルのufs故障があるときにのみ存在するだけだろう。UFSスレッドとその関連させられた待ち行列の間には１対１の対応がある。各故障は、関連する状態を持つ。

【００４０】端末状態１０４、「PANIC」はファイルシステムが修復を許されなかった、ロックできなかったか、またはUFSパニック回復アルゴリズムが内部エラーに遭遇したことのいずれかを意味する。それは、全システムに対する端末状態である。

【００４１】状態１０６「UNDEF」は、これらのレコードが新たに割り当てられて、まだ初期状態にされなかったことを意味する。それらの存在は、故障が生じたことを意味する。これらは、故障しているスレッドのコンテクストにおいて割り当てられる。

【００４２】状態１０８「INIT」は、これらのレコードが初期状態にされたことを意味する。それらの存在は、このファイルシステムがロックされる事を許され、修復できることを意味している。これらは、故障しているスレッドのコンテクストにおいて初期状態にされる。以下のUFS制御構造とポインタが、struct ufs_failureにコピーされる。

【００４３】

```
struct buf*uf_bp ;   /*スーパーブロックを含んでいるptrからbufへ*/
                     /*スーパーブロック状態をアップデートするために*/
                     /*用いられる*/
kmutex_t *uf_vfs_lockp ; /*ufsvfspを検査し修正するために用いられる*/
struct vfs_ufs *uf_vfs_vufp ; /*これが与えられたfsの複製、又は*/
                              /*オリジナルの故障であるかどうか決める*/
                              /*ために用いられる；同様にfsにつき*/
                              /*あまりに頻繁な故障を防ぐために*/
                              /*用いられる*/
struct vfs *uf_vfsp ; /*lockfsを起動し、fsにネームを得ること*/
                      /*に必要とされる構造を得るために用いられる*/
struct ufsvfs *uf_ufsvfsp ;   /*non-terminal状態において*/
                              /*関連させられたufs_failureがある時に*/
                              /*lockfsを起動し、fsがunmounted/remounted*
/
                              /*であるかどうかを決めるために*/
                              /*必要とされる構造を得るために用いられる*
/
char uf_fsname[MAXMNTLEN] ;   /*スーパーブロックからコピーされる；*/
                              /*fsがunmountedならば、エラー*/
                              /*メッセージを発生させるために用いられる*
/
char uf_panic_str[LOCKFS_MAXCOMMENTLEN] ; /*パニックに移るならarg値を*/
                                          /*含むオリジナルのパニックメッセージ*/
                                          /*を印刷するために用いられる*/
```

【００４４】状態１１０「QUEUE」は、これらのレコードがロック・スレッドのための作業待ち行列の上へ置かれたことを意味する。ロック衝突のために、いくつかのケースにおいてレコードが、作業待ち行列の上に置かれるだろうが、この状態において記録されることはできない。この場合、ロック・スレッドは、故障レコードの状態をアップデートするだろう。通常、この状態遷移は、故障スレッドのコンテクストで生じるだろう。

【００４５】端末状態１１２「REPLICA」は、UFS故障がすでに故障したファイルシステムに生じたことを意味する。ロック・スレッドは、ufsvfsの処理が必要であるので、待ち行列状態１１０からこの状態にレコードを動かすことに対して責任がある。一度記録されると、修復がすでに始められたとしてこれらのレコードは無視され、オリジナルのufs_failureと関連させられる。

【００４６】状態１１４「TRY LOCK」は、レコードがロック・スレッドによって見つけ出されて、内部のlockfs入口点（ufs fiolfs()）を起動するために必要な情報によりアップデートされたことを意味する。故障したファイルシステムのufsvfsは、この点で故障がこのファイルシステムに生じたことを示すためにアップデートされる。これは、写しを作られたUFS故障を検出するために用いられる。

【００４７】状態１１６「UMOUNT」は、ファイルシステムが修復されるよりもアンマウントされるように記録されたことを意味する。故障レコードがこの（何か他の）端末状態を成し遂げるならば、動作はとられない。

【００４８】端末状態１１８「NOT FIX」は、このファイルシステムが修復されないと記録されたこと、または修復が自動的に生じるのを妨げるために強制的にアンマウントされたことを意味する。

【００４９】状態１２０「LOCKED」は、故障したファイルシステムが成功裏にロックされたことを意味する。RPCコールは、修復を開始するためにボートに対してなされる。fsckが成功裏に始まるとき、RPCコールは、復帰する。

【００５０】状態１２２「FIXING」は、fsckが修復が進行中であることを示すスーパーブロックをアップデートしたことを意味する。

【００５１】端末状態１２４「FIXED」は、このファイルシステムがうまく修復されて、アンロックされたことを意味する。

【００５２】以下は、上に記述したいくつかの状態の間の状態遷移１００のための必要な条件である：
イベント１０２から端末状態１０４「PANIC」へ：非UFSパニックがすでに進行；ファイルシステムが定義可能でない（vp、ufsvfsp、vfspがNullである）；パニック回復の特性はこのファイルシステムの上で不可能だった；このファイルシステムに含まれる活動中のスワップファイルがある；そして、このファイルシステムに含まれる活動中のカーネル・アカウントファイルがある。

【００５３】状態１０６「UNDEF」から端末状態１０４「PANIC」へ：ファイルシステムの制御構造（ufsvfsp、vfsp、bpがstruct fsを含んでいる）が不当になった。

【００５４】状態１０８「INIT」から状態１１０「QUEUE」へ：このstruct ufs_failureは、ロック・スレッドの作業待ち行列の上に置かれた。

【００５５】状態１１０「QUEUE」から端末状態「REPLICA」へ：これはこのファイルシステムと関連させられたもう一つの活動中のUFS故障である。

【００５６】状態１１０「QUEUE」から状態１１４「TRY LOCK」へ：故障が認識されたので、ファイルシステム制御構造は変化しなかった。

【００５７】状態１１０「QUEUE」から端末状態１０４「PANIC」へ：UFS故障の数上のシステム毎の（per-system）制限が上回った；そして、UFS故障の間での最少時間のシステム毎の（per-system）制限が上回った。

【００５８】状態１１０「QUEUE」/１１４「TRY LOCK」/１２０「LOCKED」と１２２「FIXING」から端末状態１１８（「NOT FIX」）へ：このファイルシステムは、いくつかの他のスレッドによってアンマウントされた。

【００５９】状態１１４「TRY LOCK」から端末状態１０４「PANIC」へ：ファイルシステムの誤りロックをためす間にタイムアウトした；lockfsはこのファイルシステムの上でエラーEDEADLKをすることに失敗した；ブート以来このファイルシステム上にあまりに多くの故障があった；そして、この故障がこのファイルシステム上の最後の故障からあまりにすぐに生じた。

【００６０】状態１１４「TRY LOCK」から状態１１６「UMOUNT」へ：このファイルシステムは故障でアンマウントされるために記録される。

【００６１】状態１１６「UMOUNT」から端末状態「NOT FIX」へ：ファイルシステムのアンマウントが成功した。

【００６２】状態１１４「TRY LOCK」から、状態１２０「LOCKED」へ：ファイルシステムの誤りロックは成功した；そして、ファイルシステムの誤りロックが失敗した、これによりそれがすでに誤りロック中であることを示す。

【００６３】状態１２０「LOCKED」から状態１０４「PANIC」へ：fsckが開始しない又はできない。

【００６４】状態１２０「LOCKED」から状態１２２「FIXING」へ：ファイルシステムのスーパーブロックのfs_cleanフィールドが、fsckによりFSFIXへリセットされた。

【００６５】状態１２２「FIXING」から状態１２０「LOCKED」へ：ファイルシステムのスーパーブロックのfs_cleanフィールドは、fsckによりFSBADへリセットされた；そして、修復デーモンがfsckの実行中にエラーを検出した。

【００６６】状態１２２「FIXING」から端末状態１２４「FIXED」へ：ファイルシステム上の誤りロックはアンロックされた（これは、fs_clean == FSCLEANを必要とする）。

【００６７】本発明のシステムと方法を利用することを本明細書において開示したが、独立のスレッドの平均故障時間（ＭＴＴＦ）が次の故障までの時間によって増加されるので全システムの可用度が増加する。さらに、そ

れらの従属するスレッドに対するMTTFと平均修復時間（MTTR）は、任意のシステムオーバーヘッド（パニック・ダンプ、ハードウェアリセット、リブートまたは他のノードへのフェイル・オーバー（failover）のような）に対してより、むしろちょうど修復に必要な時間にまで減らされる。

【0068】本発明の原則を特定のコンピューター・オペレーティング・システムに関連して説明したが、これは単なる例示であって本発明の範囲を制限するべきものではない。

【図面の簡単な説明】

【図1】　本発明の動作環境の一部を形成している汎用計算機を代表する簡略図面である；

【図2】　図1のコンピューターの一つ以上のものから作られたコンピュータ・システムの上で、実行されるソフトウェア・アプリケーションの基礎をなしており、特に「ユーザ」、「カーネル」及び「UFS」層を、いろいろなオペレーション（例えばOp1-Op3）が動作することができる後者二つの間のVFSインタフェースとともに説明している妥当なユニックス・オペレーティング・システム層の単純化された概念上の図解である；

【図3】　内部のデータ構造が正常動作の間、UFS故障の前に変化する方法の単純化された概念上の表示である；

【図4】　認識されていて、他のUFSスレッドのVFS層でのブロッキングに終わっているUFS故障の概念上の表示である；

【図5】　システムとファイルシステム・パニック回復の方法の単純化された概念上の表示であり、ここでシステムは、アンマウントまたはロックされた条件に留められることができ、そして、ロック・スレッドは、ファイルシステムのチェッカーを実行するために修復を開始するように進む；

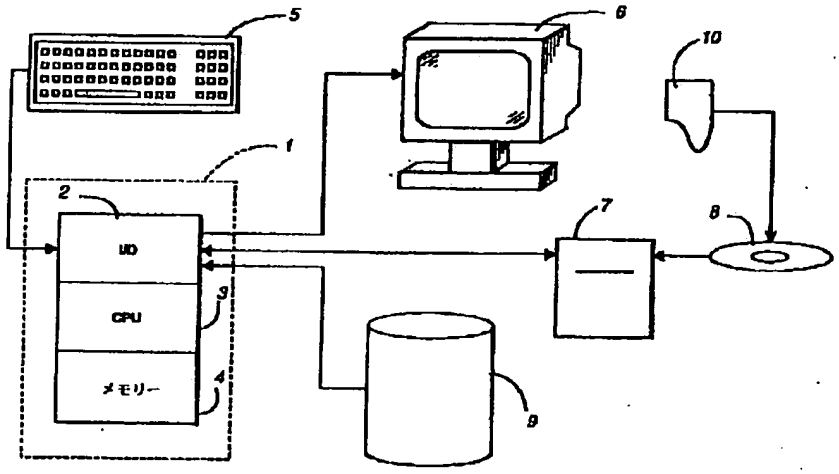
【図6】　以前の図において例を示した、層の単純化された概念上の表示であり、ここでファイルシステムのチェック「fsck」が修復を完了し、他のスレッドのオペレーションの続行が許される；

【図7】　本明細書において開示したファイルシステム・パニック回復のシステム及び方法の特定のインプリメンテーションに従うUFS故障のイベントにおける複数の状態の間でできる遷移を図示している典型的な状態図である。

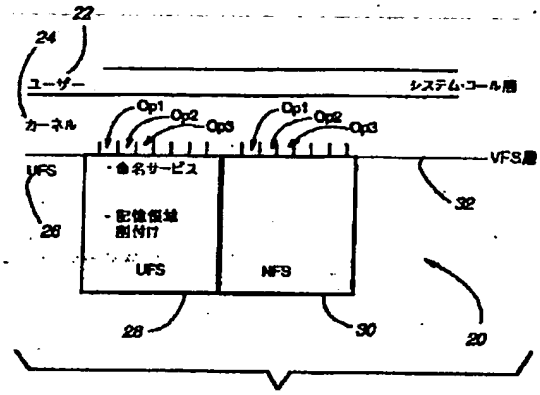
【符号の説明】

1：プロセッサー
2：入出力セクション
3：中央処理装置
4：メモリー・セクション
5：キーボード
6：表示装置
7：CD－ROM装置
8：CD－ROM媒体
9：磁気ディスク装置
10：プログラム
20：ユニックスOSシステム層
36：誤りロック
38、54：UFSスレッド
40：記憶装置
42：故障スレッド
80：他のスレッド

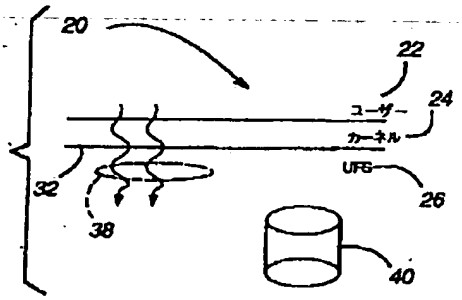
【図1】



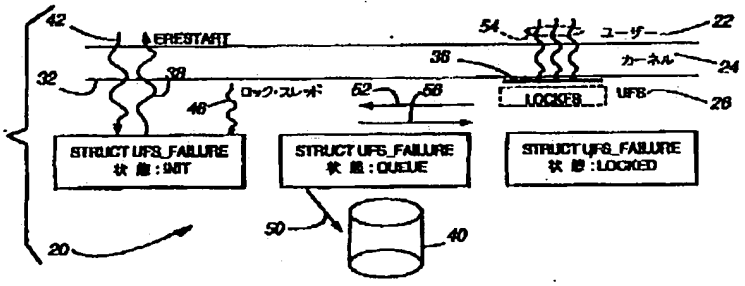
【図2】


22
24
ユーザー
システム・コール層
カーネル
Op1
Op2
Op3
Op1
Op2
Op3
VFS層
UFS
・命名サービス
・記憶領域
割付け
UFS
NFS
26
32
28
30
20

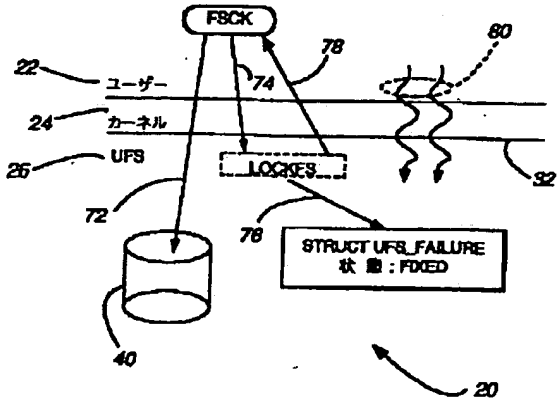
【図3】


20
22
ユーザー
24
カーネル
UFS
26
32
38
40

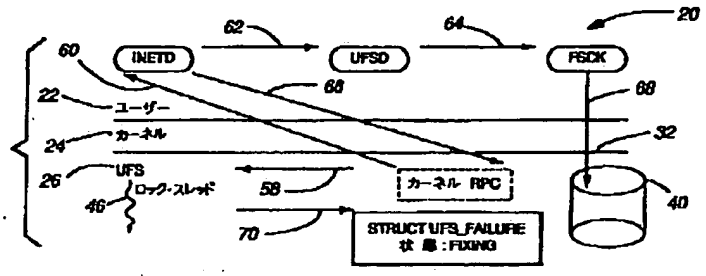
【図4】


42
ERESTART
54
ユーザー
22
36
カーネル
24
32
38
ロック・スレッド
52
56
LOCKES
UFS
26
46
STRUCT UFS_FAILURE
状 態 : WAIT
STRUCT UFS_FAILURE
状 態 : QUEUE
STRUCT UFS_FAILURE
状 態 : LOCKED
50
40
20

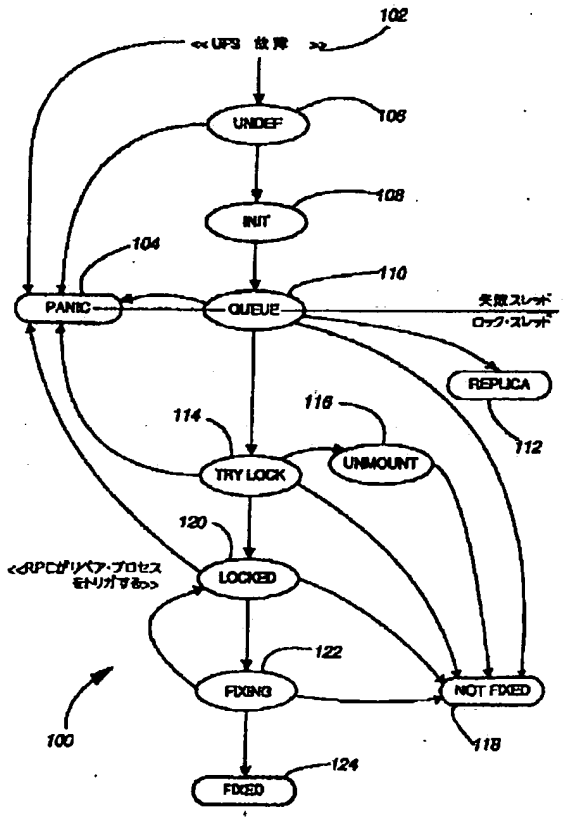
【図6】


FSCK
78
80
74
22
ユーザー
24
カーネル
UFS
26
LOCKES
32
72
76
STRUCT UFS_FAILURE
状 態 : FIXED
40
20

【図5】


INETD
UFSD
FSCK
ユーザー
カーネル
UFS
ロック・スレッド
カーネル RPC
STRUCT UFS_FAILURE
状 態 : FIXING
20
22
24
26
32
40
46
58
60
62
64
68
70

【図7】


<< UFS 故障 >>
UNDEF
INIT
PANIC
QUEUE
失敗スレッド
ロック・スレッド
REPLICA
TRY LOCK
UNMOUNT
<<RPCにリペア・プロセス
をトリガする>>
LOCKED
FIXING
NOT FIXED
FIXED
100
102
104
108
110
112
114
116
118
120
122
124